

Design and ARM-embedded implementation of a chaotic map-based multicast scheme for multiuser speech wireless communication

Qiuye Gan¹, Simin Yu¹, Chengqing Li², Jinhu Lü³, Zhuosheng Lin¹, and Ping Chen¹

¹College of Automation, Guangdong University of Technology, Guangzhou, China (siminyu@163.com)

²College of Information Engineering, Xiangtan University, Xiangtan 411105, Hunan, China
(DrChengqingLi@gmail.com)

³Academy of Mathematics and Systems Sciences, Chinese Academy of Sciences, Beijing 100190, China
(jhlul@iss.ac.cn)

SUMMARY

This paper proposes a chaotic map-based multicast scheme for multiuser speech wireless communication and implements it in an ARM platform. The scheme compresses the digital audio signal decoded by a sound card and then encrypts it with a three-level chaotic encryption scheme. First, the position of every bit of the compressed data is permuted randomly with a pseudo-random number sequence (PRNS) generated by a 6-D chaotic map. Then, the obtained data are further permuted in the level of byte with a PRNS generated by a 7-D chaotic map. Finally, it is operated with a multi-round chaotic stream cipher. The whole system owns the following merits: the redundancy in the original audio file is reduced effectively and the corresponding unicity distance is increased; the balancing point between a high security level of the system and real-time conduction speed is achieved well. In the ARM implementation, the framework of communication of multicast multiuser in a subnet and the Internet Group Manage Protocol is adopted to obtain the function of communication between one client and other ones. Comprehensive test results were provided to show the feasibility and security performance of the whole system.

keywords and phrases: ARM-embedded implementation, multicast-multiuser, chaotic map, secure communication, speech, WIFI.

1 Introduction

With popularization of speech (voice and audio) record devices and development of network transmission techniques, the security of speech transmission is becoming more and more important. To cope with the challenge, a great number of encryption schemes based on chaos were proposed to protect speech data. It is well known that some significant properties of chaos, such as positive Lyapunov exponents, ergodicity, quasi-randomness, sensitive dependence on initial conditions, and system parameters, have suggested chaotic dynamics being a promising alternative for conventional cryptographic algorithm [1–3]. The function of chaos used in encryption can be categorized as the following classes: 1) generation of scrambling (permutation) relation [4, 5]; 2) producing pseudo-random number sequence (PRNS) [6, 7]; 3) realization of secure communication via chaos synchronization [8–10]; 4) construction of public encryption scheme [11]. Meanwhile, some cryptanalytic works pointed out that some speech encryption schemes are insecure against the corresponding specific attack methods [12, 13]. To counteract dynamical degradation of digital chaotic systems, various methods were proposed to weaken or even eliminate the negative influence caused by the digitization [14–16]. Some methods based on complex networks were proposed to measure the real degradation degree of the counteract measures [16, 17].

Scrambling is one of most efficient and simple encryption methods [18, 19]. Although reference [20] presents optimal quantitative cryptanalysis of any position permutation-only encryption scheme against known/chosen-plaintext attack, they are still widely used in the design of multimedia encryption schemes. As size of speech and video data is much larger than that of image data, only some important parameters are selected to be encrypted [21, 22]. To evaluate their importance degree, various criteria were proposed, such as sensible influence [23, 24], and decoding importance [25]. In the field of multimedia secure communication, most efforts were spent on software implementation, only few works touched hardware implementation of multimedia encryption scheme [26].

In this paper, we propose a novel TCP-based multicast scheme for multiuser speech wireless communication, and implement it in ARM hardware platform. The scheme is designed for scenario of wireless communication where the balancing point between sufficient security level and low computation load is expected. The kernel of the scheme is to compress original speech data input from Audio Codec based on the adaptive differential pulse code modulation (ADPCM) algorithm designed by Interactive Multimedia Association (IMA) [27]. Then, it encrypts the compressed data with the following three levels of chaotic encryption: scrambling 1-bit element in each 1-byte data with a 6-D chaotic map; scrambling position of every 1-byte data with a 7-D chaotic map; encrypting compressed speech data with a chaotic stream cipher multiple times. The encryption kernel is cascaded with a multicast method connecting sender point to receiver point with a way of point-to-multiple. Security performance of the proposed system is verified from various aspects: NIST test, statistical analysis, sensitivity with respect to change of secret key and size estimation of secret key space. Compared with video encryption scheme proposed in [26], this scheme still run sufficient fast as size of speech data is much smaller than that of video data of the same playing time.

The remainder of this paper is organized as follows. In Sec. 2, a brief introduction is given to multicast-multiuser wireless communication system. Section 3 presents basic principle of speech compression. Section 4 proposes a discrete-time chaotic system without any degeneration and uses it to implement secure and efficient speech communication scheme. Section 5 designs a multi-round chaotic stream cipher for speech communication. Section 6 further discusses cascading the proposed speech communication in the multicast environment for multi-user wireless communication. Section 7 analyzes security performance of the whole secure communication system. Finally, the last section concludes the paper.

2 Multicast-multiuser wireless communication system

The function diagram of the ARM-based multicast-multiuser communication system is shown in Fig. 1, where both the sender and the receiver use the same development board ARM9Tiny4412. Its hardware part includes the following key components: four-core Cortex-A9 processing unit of dominant frequency 1.5GHz, sound card, wireless network card, 1GB DDR3 SDRAM (32-bit data bus), 4GB Flash, and USB port. The software system can be partitioned into the driver layer and application layer. The former mainly includes sound card driver, network card driver, the Linux kernel and so on. The latter includes adaptive speech acquisition, speech compression, three-level chaotic encryption, the sender of encrypted speech signal, the receiver for encrypted speech signal, three-level chaotic decryption, speech adaptive decompression, and speech broadcaster.

As shown in Fig. 2, the ARM-based multicast-multiuser communication system owns the following features. In the multicast architecture based on ARM+linux embedded devices, a multicast-multiuser chaotic map-based speech communication is implemented. In the uplink communication of both the sending end and receiving end, the received data is sent to the application layer through the sound card, network card and Linux kernel. In the downlink communication, the data is sent to the sound card and network card by them also. All the receivers can receive multicast data via WIFI signal if they registered the multicast membership. When the electricity of ARM is turned on, its own WIFI connection program is run first [28]. Once any WIFI signal is detected, WIFI is connected successfully by entering the right account number and password.

3 Design of data acquisition and compression scheme

Redundancy existing in raw message is very useful for error detection and correction in communication. However, it is very harmful in cryptology. As pointed out by Shannon, redundancy provides basis for cryptanalysis, which can be represented as a formula $U = H/D$, where U , H and D denote unicity distance, entropy of secret key, and redundancy degree, respectively. It means that there is sole solution for deciphering an encryption scheme when the number of ciphertext characters intercepted by the cryptanalysts is greater than the unicity distance. Otherwise, there is multiple solutions and the complexity of the deciphering is very large. From such viewpoint, one can see that compressing speech data before encryption is helpful for improving security performance of the whole secure communication system.

Every 16-bit digital speech data input from the sound card is compressed into 4-bit data through IMA-ADPCM, whose most significant bit is the sign bit. Since the C language does not support the data types of 4 bits, two variables of char type are used to represent one compressed variable. Concretely, the PCM data is compressed and stored into buffer in chronological order, and each BYTE space store two compressed data, where the four least significant bits store the first one and the four most significant bits store the second one.

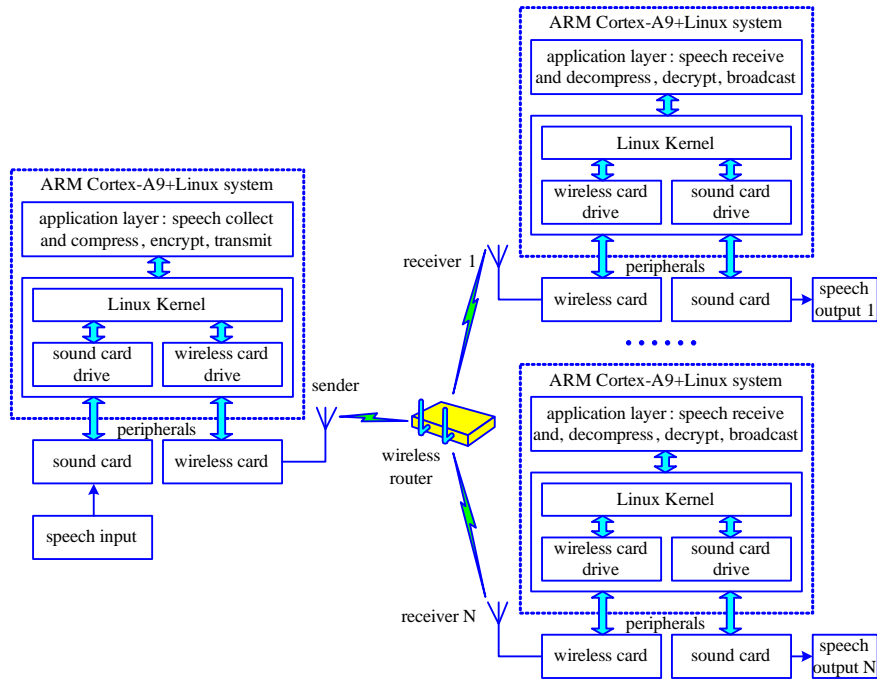


Figure 1: The function diagram of the ARM-based multicast-multuser communication system.

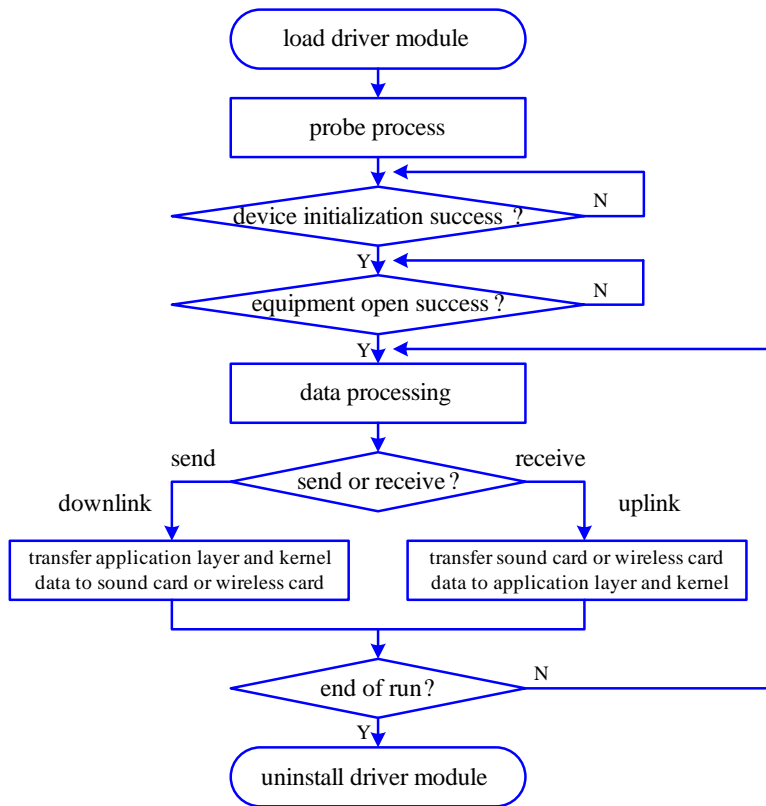


Figure 2: Flow chart of uplink/downlink communication.

The schematic of speech compression algorithm based on IMA-ADPCM is shown in Fig. 3. First, the 32768 16-bit data collected by sound card are stored into inbuf, and then compressed into 32768 4-bit data by IMA-ADPCM algorithm. Every two adjacent 4-bit data are combined into one 8-bit data. Finally, 16384 8-bit compressed data are stored into outbuf as one frame.

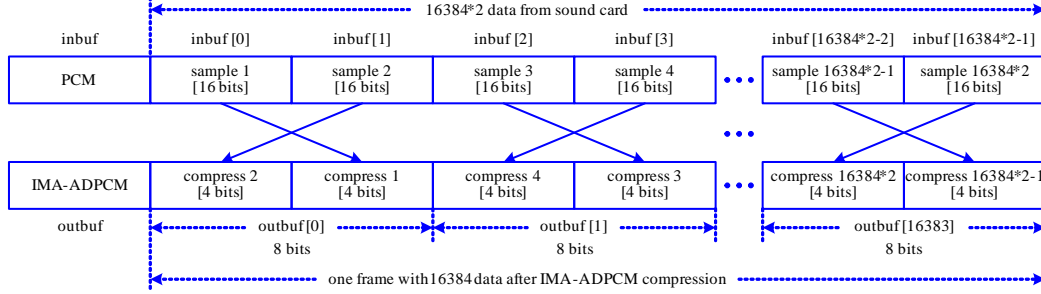


Figure 3: Schematic diagram of the speech compression based on IMA-ADPCM.

4 Design of scrambling scheme for compressed speech data

This section discusses the design of scrambling scheme for compressed speech data. First, position of every bit of the compressed speech data is permuted randomly with a pseudo-random number sequence (PRNS) generated by a 6-D chaotic map. Then, the scrambled data is further permuted with a PRNS generated by a 7-D chaotic map in the level of byte.

4.1 Construction of an n -dimensional discrete-time chaotic system

Like [26], we adopt the idea of Chen-Lai chaotification scheme in [2] to construct an n -dimensional discrete-time chaotic system, which can be generally presented as

$$\begin{cases} x_{1,k+1} = \text{mod}(A_{11}x_{1,k} + A_{12}x_{2,k} + \dots + A_{1n}x_{n,k}, 2^{N_1}), \\ x_{2,k+1} = \text{mod}(A_{21}x_{1,k} + A_{22}x_{2,k} + \dots + A_{2n}x_{n,k}, 2^{N_2}), \\ \dots \\ x_{n,k+1} = \text{mod}(A_{n1}x_{1,k} + A_{n2}x_{2,k} + \dots + A_{nn}x_{n,k}, 2^{N_n}), \end{cases} \quad (1)$$

where $\text{mod}(m, n) = m - n \cdot \lfloor \frac{m}{n} \rfloor$, and $\lfloor x \rfloor$ gives the largest integer less than or equal to x . In the proposed secure communication system, we set $N_1 = N_2 = \dots = N_n = N_0$, and $N = 2^{N_1} \cdot 2^{N_2} \dots 2^{N_n} = 2^{nN_0}$.

The coefficients A_{ij} in system (1) are derived from matrix

$$\mathbf{A} = \begin{bmatrix} A_{11} & A_{12} & \dots & A_{1n} \\ A_{21} & A_{22} & \dots & A_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ A_{n1} & A_{n2} & \dots & A_{nn} \end{bmatrix} = \prod_{i=1}^{n-1} \left(\prod_{j=i+1}^n \mathbf{T}_{ij} \right), \quad (2)$$

where \mathbf{T}_{ij} is given by

$$\mathbf{T}_{ij} = \begin{pmatrix} 1 & 0 & \cdots & & & \\ 0 & 1 & \cdots & & & \\ \vdots & \vdots & \ddots & & & \\ & & & A_{i,i}^{(11)} & 0 & \cdots & 0 & \cdots & 0 & A_{i,j}^{(12)} \\ & & & 0 & 1 & \cdots & 0 & \cdots & 0 & 0 \\ & & & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{0} & & & 0 & 0 & \cdots & 1 & \cdots & 0 & 0 & \mathbf{0} \\ & & & \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \vdots & \\ & & & 0 & 0 & \cdots & 0 & \cdots & 1 & 0 \\ & & & A_{j,i}^{(21)} & 0 & \cdots & 0 & \cdots & 0 & A_{j,j}^{(22)} \\ & & & & & & & \ddots & \cdots & \cdots \\ 0 & & & \mathbf{0} & & & & \vdots & 1 & 0 \\ & & & & & & & \vdots & 0 & 1 \end{pmatrix}. \quad (3)$$

Algorithm 1 The Transform function

```

1: function TRANSFORM(A)
2:    $S = 0$ 
3:   for  $x_{1,k} \leftarrow 0, 2^{N_0} - 1$  do
4:     for  $x_{2,k} \leftarrow 0, 2^{N_0} - 1$  do ...
5:       for  $x_{n,k} \leftarrow 0, 2^{N_0} - 1$  do
6:          $E(S, 1) = \text{mod}(A_{11}x_{1,k} + A_{12}x_{2,k} + \cdots + A_{1n}x_{n,k}, 2^{N_0})$ 
7:          $E(S, 2) = \text{mod}(A_{21}x_{1,k} + A_{22}x_{2,k} + \cdots + A_{2n}x_{n,k}, 2^{N_0})$ 
8:         ...
9:          $E(S, n) = \text{mod}(A_{n1}x_{1,k} + A_{n2}x_{2,k} + \cdots + A_{nn}x_{n,k}, 2^{N_0})$ 
10:         $S = S + 1$ 
11:      end for ...
12:    end for
13:  end for
14:  return  $E$ 
15: end function

```

The mathematical expression of the serial number is

$$S = \sum_{i=1}^{nN_0/2} x_{i,k} \cdot 2^{n \cdot N_0 - 2 \cdot i}. \quad (4)$$

Utilizing Algorithm 1, the corresponding scrambling index can be obtained via

$$E(S) = \sum_{i=1}^{nN_0/2} E(S, \alpha_i) \cdot 2^{n \cdot N_0 - 2 \cdot i}, \quad (5)$$

where $\alpha_i \in \{1, 2, \dots, n\}$, and $\alpha_1 \neq \alpha_2 \neq \dots \neq \alpha_n$.

4.2 Scrambling the compressed speech data in the level of bit

Set the variables in system (1) as $n = 6$, $N_0 = 2$, the number of possible scrambling is $N = 2^{n \cdot N_0} = 2^{6 \cdot 2} = 4096$. Then, position of every bit of the compressed data is permuted randomly with a PRNS generated by the 6-D chaotic map constructed in the above sub-section. As illustrated in Fig. 3, each frame of compressed speech data has 16384 bits. So, it needs to be divided into 32 groups, and each group has 512 bytes. There are $512 \cdot 8 = 4096$ bits of data, so the positions of the 4096 bits of data are scrambled to complete the encryption. After repeating the method 32 times to complete encryption of one frame of compressed speech data, the diagram of scrambling 1-bit data position is shown in Fig. 4. Similarly, the 8-bit data in $\text{inbuf}[k_4]$ becomes $\text{outbuf}[k_4]$ by scrambling 1-bit data position 8 times, which is shown in Fig. 5. The original speech

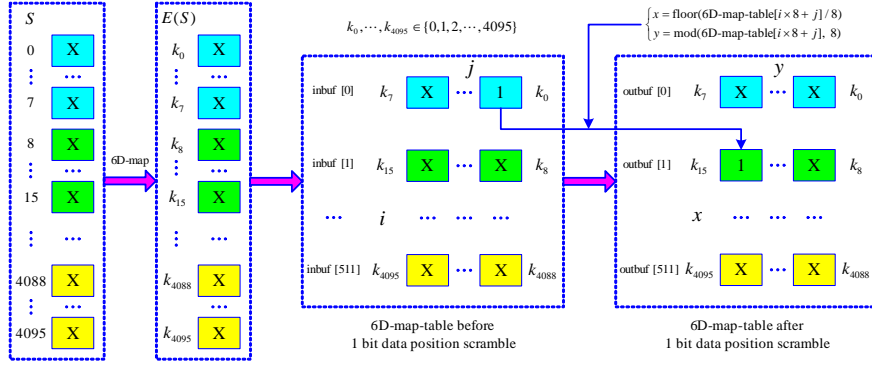


Figure 4: Schematic diagram of bit permutation.

compressed data is temporarily stored in the inbuf, the encrypted data is stored in the outbuf, and all bits of data in the outbuf are set as zero initially.

According to Fig. 4, the corresponding chaotic encryption algorithm is described as follows:

- *Step 1:* According to Eq. (4), Eq. (5) and Fig. 4, the value of S is mapped to $E(S)$, and the corresponding coordinate is (i, j) ;
- *Step 2:* Extract 1-bit data unit corresponding to the i -th row and the j -th column in inbuf as

$$(\text{inbuf}[i] \gg j) \& 1 \in \{0, 1\}.$$

- *Step 3:* The positions of data are then scrambled by 6-D chaotic mapping table, and the coordinates are mapped to

$$\begin{cases} x = \lfloor 6\text{-D-map-table}[i \cdot 8 + j] / 8 \rfloor, \\ y = \text{mod}(6\text{-D-map-table}[i \cdot 8 + j], 8), \end{cases} \quad (6)$$

- *Step 4:* Replace 1-bit data in the corresponding unit from (i, j) to (x, y) via

$$\text{Outbuf}[x] = (((\text{Inbuf}[i] \gg j) \& 1) \ll y) \vee \text{Outbuf}[x],$$

where the symbol “ $\gg j$ ” denotes right shift of j bits, “ $\ll y$ ” denotes left shift of y bits, “ $\&$ ” denotes the bitand operation, “ \vee ” denotes the bitwise OR operation, and “mod” represents modulo operation.

Set the parameters in Eq. (3), $A_{i,i}^{(11)} = A^{(11)} = 1$, $A_{i,j}^{(12)} = A^{(12)} = 1$, $A_{j,i}^{(21)} = A^{(21)} = 1$, $A_{j,j}^{(22)} = A^{(22)} = 2$, $n = 6$, then one has

$$\mathbf{A}_6 = \prod_{i=1}^5 \left(\prod_{j=i+1}^6 \mathbf{T}_{ij} \right) = \begin{pmatrix} 1 & 5 & 18 & 52 & 121 & 197 \\ 1 & 6 & 22 & 64 & 149 & 242 \\ 1 & 5 & 19 & 55 & 128 & 208 \\ 1 & 4 & 13 & 38 & 88 & 144 \\ 1 & 3 & 8 & 20 & 48 & 80 \\ 1 & 2 & 4 & 8 & 16 & 32 \end{pmatrix}. \quad (7)$$

A concrete example is provided to illustrate the algorithm for scrambling of 1-bit data position:

- According to Eq. (4), Eq. (5) and Fig. 4, when $S = 14$, $E(S) = 1792$, then the corresponding coordinate $(i, j) = (1, 6)$.
- Extract 1-bit data unit corresponding to the $i = 1$ -th row and the $j = 6$ -th column in inbuf as

$$(\text{inbuf}[1] \gg 6) \& 1 = 0.$$

- The positions of data are then scrambled by 6-D chaotic mapping table, and the coordinates are mapped to

$$\begin{cases} x = \lfloor 6\text{-D-map-table}[1 \cdot 8 + 6] / 8 \rfloor = \lfloor 1792 / 8 \rfloor = 224, \\ y = \text{mod}(6\text{-D-map-table}[1 \cdot 8 + 6], 8) \\ = \text{mod}(1792, 8) = 0. \end{cases}$$

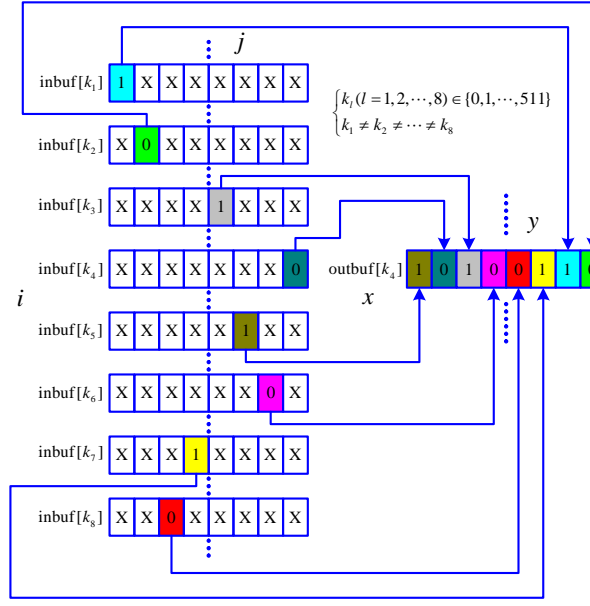


Figure 5: Schematic diagram of eight rounds of bit permutation.

- The 1-bit data in the corresponding unit is moved from $(i, j) = (1, 6)$ to $(x, y) = (224, 0)$ via

$$\text{Outbuf}[224] = (((\text{Inbuf}[1] \gg 6) \& 1) \ll 0) \vee \text{Outbuf}[224].$$

Finally, all the 0's in $(i, j) = (1, 6)$ are replaced by the corresponding unit in $(x, y) = (224, 0)$. The rest cases follow the rule shown in Fig. 6 similarly.

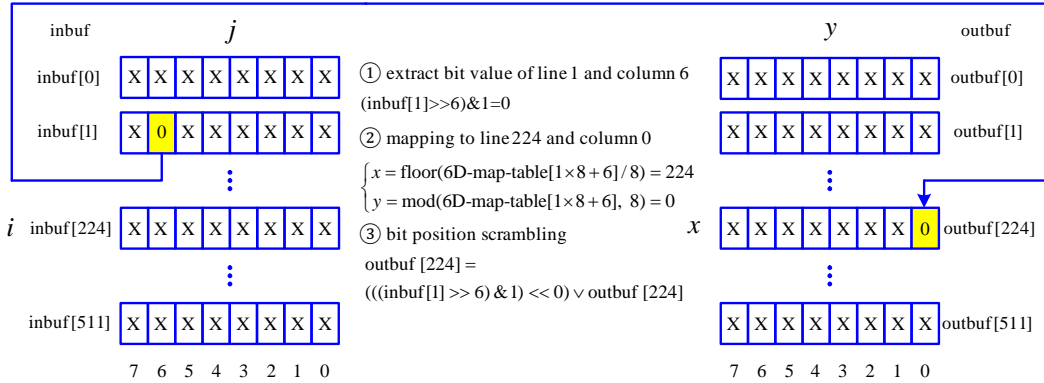


Figure 6: The schematic diagram of scrambling one typical element.

To decrypt the received messages, two cases must be distinguished: communication with matched parameters and that with mismatched parameters. Taking the order of 15 matrices in Eq. (7) as the key, the analysis results with matched parameters and that with mismatched parameters are shown as follows:

1) With matched parameters, the permutation matrix at the receiving end is still the same as that shown in Eq. (7). When $(x, y) = (224, 0)$, one has $(8x + y) = (8 \cdot 224 + 0) = 1792$. After the decryption, 0's can be replaced back to the correct position $(i, j) = (1, 6)$. Similarly, the other encrypted information can be correctly decrypted at the receiving end.

2) Assume that the original arrangement is

$$\mathbf{T}_{12} \mathbf{T}_{13} \mathbf{T}_{14} \mathbf{T}_{15} \mathbf{T}_{16} \mathbf{T}_{23} \mathbf{T}_{24} \mathbf{T}_{25} \mathbf{T}_{26} \mathbf{T}_{34} \mathbf{T}_{35} \mathbf{T}_{36} \mathbf{T}_{45} \mathbf{T}_{46} \mathbf{T}_{56},$$

\mathbf{T}_{12} and \mathbf{T}_{34} are swapped by the mismatched parameters, the transformation matrix become

$$\mathbf{A}_6 = \begin{pmatrix} 5 & 9 & 11 & 29 & 68 & 113 \\ 1 & 2 & 3 & 8 & 19 & 31 \\ 11 & 20 & 24 & 65 & 152 & 252 \\ 16 & 29 & 33 & 91 & 212 & 352 \\ 4 & 7 & 8 & 20 & 48 & 80 \\ 3 & 5 & 4 & 8 & 16 & 32 \end{pmatrix}.$$

As $8x + y = 8 \cdot 224 + 0 = 1792$, 0's is replaced by the decryption process to a wrong position, $(i, j) = (0, 1)$, instead of right position $(i, j) = (1, 6)$. So, the encrypted information cannot be correctly decrypted at the receiving end.

4.3 Scrambling the compressed speech data in the level of BYTE

In the second-level of encryption, let $n = 7$, $N_0 = 2$, the number of the position scrambling is $N = 2^{n \cdot N_0} = 2^{7 \cdot 2} = 16384$. The data is further permuted in the level of byte with a PRNS generated by a 7-D chaotic map. Positions of frames of size 1×16384 are then scrambled as shown in Fig. 7, where the original speech compressed data is temporarily stored in the inbuf, the encrypted data is temporarily stored in the outbuf, where $k_i \in \{0, 1, 2, \dots, 16383\}$ and $k_0 \neq k_1 \neq k_2 \neq \dots \neq k_{16383}$.

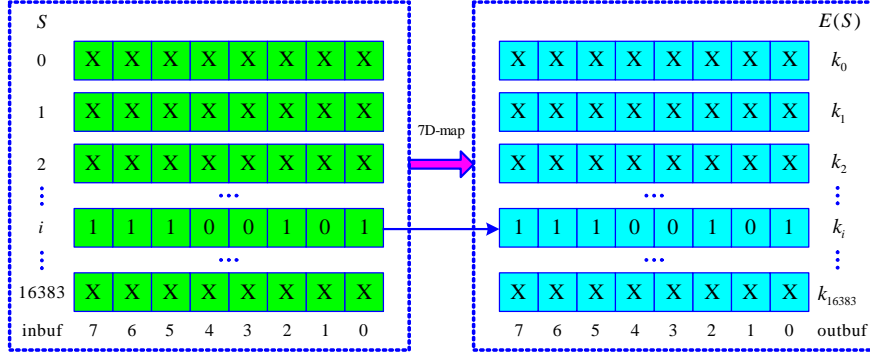


Figure 7: Schematic diagram of scrambling and anti-scrambling of 8 bits speech compressed data.

When $n = 7$, the sequence number S of the speech compression data before encryption is

$$S = \sum_{i=1}^7 x_{i,k} \cdot 2^{14-2i}. \quad (8)$$

The corresponding sequence number S of the speech compression data after encryption is

$$E(S) = \sum_{i=1}^7 E(S, \alpha_i) \cdot 2^{14-2i}, \quad (9)$$

where $\alpha_i \in \{1, 2, \dots, 7\}$, $\alpha_1 \neq \alpha_2 \dots \neq \alpha_7$.

Setting coefficients in Eq. (3), $A_{i,i}^{(11)} = A^{(11)} = 1$, $A_{i,j}^{(12)} = A^{(12)} = 1$, $A_{j,i}^{(21)} = A^{(21)} = 1$, $A_{j,j}^{(22)} = A^{(22)} = 2$, $n = 7$, one has

$$\mathbf{A}_7 = \prod_{i=1}^6 \left(\prod_{j=i+1}^7 \mathbf{T}_{ij} \right) \quad (10)$$

$$= \begin{pmatrix} 1 & 6 & 25 & 84 & 237 & 550 & 903 \\ 1 & 7 & 30 & 102 & 289 & 671 & 1100 \\ 1 & 6 & 26 & 88 & 249 & 578 & 948 \\ 1 & 5 & 19 & 63 & 176 & 408 & 672 \\ 1 & 4 & 13 & 38 & 104 & 240 & 400 \\ 1 & 3 & 8 & 20 & 48 & 112 & 192 \\ 1 & 2 & 4 & 8 & 16 & 32 & 64 \end{pmatrix}.$$

A concrete example is illustrated the algorithm for scrambling position of every byte. Let $S = i = 100$, $E(S) = E(i) = k_i = 3296$. The position $S = 100$ for 8-bit data “11100101” is replaced by $E(S) = 3296$.

Taking the order of 21 matrices in Eq. (10) as the key, the analysis results with matched parameters and that with mismatched parameters are shown as follows.

1) With matched parameters, the permutation matrix at the receiving end is still that shown in Eq. (10). Given $E(S) = 3296$, one can get $S = 100$. After the decryption process, the 8-bit data is replaced back to the right position. The other cases run in the similar way. So, the encrypted speech data can be decrypted successfully.

2) Assume the original arrangement is

$$\mathbf{T}_{12}\mathbf{T}_{13}\mathbf{T}_{14}\mathbf{T}_{15}\mathbf{T}_{16}\mathbf{T}_{17}\mathbf{T}_{23}\mathbf{T}_{24}\mathbf{T}_{25}\mathbf{T}_{26}\mathbf{T}_{27}\mathbf{T}_{34}\mathbf{T}_{35}\mathbf{T}_{36}\mathbf{T}_{37}\mathbf{T}_{45}\mathbf{T}_{46}\mathbf{T}_{47}\mathbf{T}_{56}\mathbf{T}_{57}\mathbf{T}_{67},$$

\mathbf{T}_{12} and \mathbf{T}_{45} are swapped by mismatched parameters, the transformation matrix becomes

$$A_7 = \begin{pmatrix} 6 & 11 & 20 & 53 & 138 & 322 & 539 \\ 1 & 2 & 5 & 14 & 37 & 87 & 144 \\ 7 & 13 & 26 & 70 & 183 & 428 & 714 \\ 11 & 20 & 32 & 81 & 212 & 492 & 828 \\ 16 & 29 & 45 & 111 & 292 & 676 & 1140 \\ 4 & 7 & 8 & 20 & 48 & 112 & 192 \\ 3 & 5 & 4 & 8 & 16 & 32 & 64 \end{pmatrix}.$$

Setting the above matrix and Eq. (8), (9), $S = 491$, one can get $E(S) = 3296$. After the decryption, the byte is replaced to the wrong position of index $S = 491$ instead of index $S = 100$. So, the encrypted information cannot be correctly decrypted at the receiving end.

5 Encrypting compressed speech data by multi-round stream cipher

This section discusses design of the third level of chaotic encryption scheme for the compressed speech data. First, we propose a general design method for discrete chaotic systems without degeneration of positive Lyapunov exponent. Then, we use 3-D version of the chaotic systems to encrypt 8-bit compressed speech data multiple times.

5.1 Design of discrete time chaotic system without degeneration of positive Lyapunov exponents

The relation between random statistical characteristic of a chaotic system and Kolmogorov-Sinai entropy can be represented as

$$h_{KS} = \sum_i LE_i^+,$$

where LE_i^+ is the number of positive Lyapunov exponents [2]. As the larger value of h_{KS} means the better statistical properties of randomness, a chaotic system owns better random statistical performance if it has more and larger positive Lyapunov exponents. In our design of discrete time chaotic system, we try to make the number and the quantity of its positive Lyapunov exponents both reach maximum. The concrete approaches can be described as follows.

Step 1: The general form of a nominal system with asymptotically stable origin is

$$\mathbf{x}(k+1) = \mathbf{C}\mathbf{x}(k), \quad (11)$$

where $\mathbf{C} = \begin{pmatrix} c_{11} & \cdots & c_{1n} \\ \vdots & \ddots & \vdots \\ c_{n1} & \cdots & c_{nn} \end{pmatrix}$ is a constant matrix, and all characteristic roots of the nominal system are located inside an unit circle.

Step 2: To control the nominal system (11) effectively, transform it as

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k), \quad (12)$$

where

$$\mathbf{A} = \mathbf{PCP}^{-1},$$

and \mathbf{P} is a nonsingular matrix. Due to the properties of similarity transformation, the two systems, (11) and (12), have the same characteristic roots and stability performance.

Step 3: Control the system (12) with the strategy of anticontrol proposed in [2] and get a uniformly bounded system

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{g}(\sigma\mathbf{x}(k), \varepsilon), \quad (13)$$

where $\mathbf{g}(\sigma\mathbf{x}(k), \varepsilon)$ is a uniformly bounded controller and \mathbf{B} is a control matrix.

Step 4: Exert pole assignment on the system (13) by adjusting \mathbf{B} , σ and ε and make the number of its positive Lyapunov exponents L reach maximum and the quantities can be sufficiently large, i.e. the system (13) reach the state without degeneration.

To illustrate performance of the above method, we present a concrete example of 3-D system,

$$\begin{cases} x_1(k+1) = a_{11}x_1(k) + a_{12}x_2(k) + a_{13}x_3(k) \triangleq f_1(\cdot), \\ x_2(k+1) = a_{21}x_1(k) + a_{22}x_2(k) + a_{23}x_3(k) \triangleq f_2(\cdot), \\ x_3(k+1) = a_{31}x_1(k) + a_{32}x_2(k) + a_{33}x_3(k) \\ \quad + \varepsilon \sin(\sigma x_1(k)) \triangleq f_3(\cdot), \end{cases} \quad (14)$$

where

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} = \begin{pmatrix} 0.205 & -0.595 & 0.265 \\ -0.265 & -0.125 & 0.595 \\ 0.33 & -0.33 & 0.47 \end{pmatrix},$$

and $\sigma = 6.6667\varepsilon/10^{-4}$.

Figure 8 depicts Lyapunov spectrum of the system (14) for $\varepsilon \in [0, 300]$, where LE_1 , LE_2 and LE_3 denote Lyapunov exponents in three dimensions, respectively. From Fig. 8, we can see that the Lyapunov exponent becomes more and more large and be positive in most cases when $\varepsilon > 100$. When $\varepsilon = 3 \cdot 10^8$, $\sigma = 2 \cdot 10^5$, one can calculate $LE_1 = 14.9$, $LE_2 = 14.8$, $LE_3 = 0.19$, the corresponding phase diagram is shown in Fig. 9.

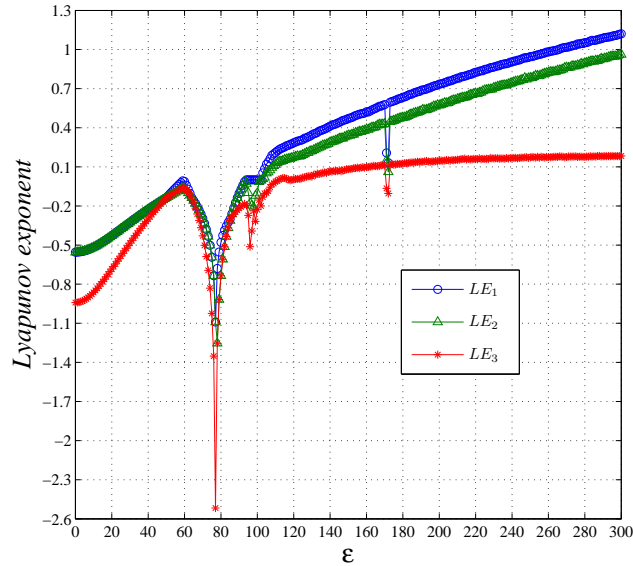


Figure 8: Lyapunov spectrum with parameter ε and σ .

5.2 The design principle of multi-round stream cipher

The design principle of multi-round stream cipher is shown in Fig. 10. The iterative equations $x_i^{(d)}(k+1) = f_i^{(d)}(\cdot)$ and $x_i^{(r)}(k+1) = f_i^{(r)}(\cdot)$ ($i = 1, 2, 3$) defined in system (14), are implemented with double-precision

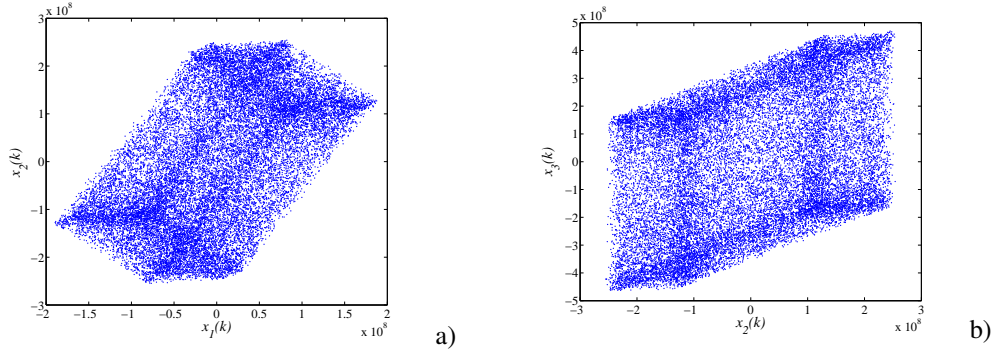


Figure 9: Phase diagram of a 3-D chaotic attractor: a) $x_1 - x_2$ plane; b) $x_2 - x_3$ plane.

arithmetic. Functions $\text{int}[x_1^{(d)}(k)]$ and $\text{int}[x_1^{(r)}(k)]$ transform the variables from double type to 64-bit integer. The modulo function $\text{mod}(2^8)$ get the eight least significant bits of the 64-bit integer, which will be used for encryption of 8-bit speech signal. To further enhance the ability to resist differential attack, we propose an encryption scheme by M -rounds chaotic stream cipher for each frame of the speech data. Security analysis shows that this scheme can greatly enhance the ability to resist differential attack. The whole framework shown in Fig. 10 consists of four parts: encryption scheme based on multi-round of chaotic stream cipher at the sending end, decryption scheme based on multi-round of chaotic stream cipher at the receiving end, channel and the switch function of K_1 , K_2 , K_3 , and K_4 . The four switch functions can be programmed through C language. The main feature of this scheme is that the $x_1^{(d)}(k)$ at second and third equations of chaotic system in Fig. 10 are replaced by the output feedback through closed-loop feedback (including the speech signal) at the sending end. Then, the final encrypted speech signal after M -round encryption is transmitted via WIFI. At the receiving end, $x_1^{(r)}(k)$ at second and third equations of chaotic system are replaced by the received encrypted speech signal after M -round encryption. Then, the corresponding M -rounds of speech signal decryption makes chaotic systems at the sending end and the receiving end synchronize, i.e., the decryption can be achieved successfully.

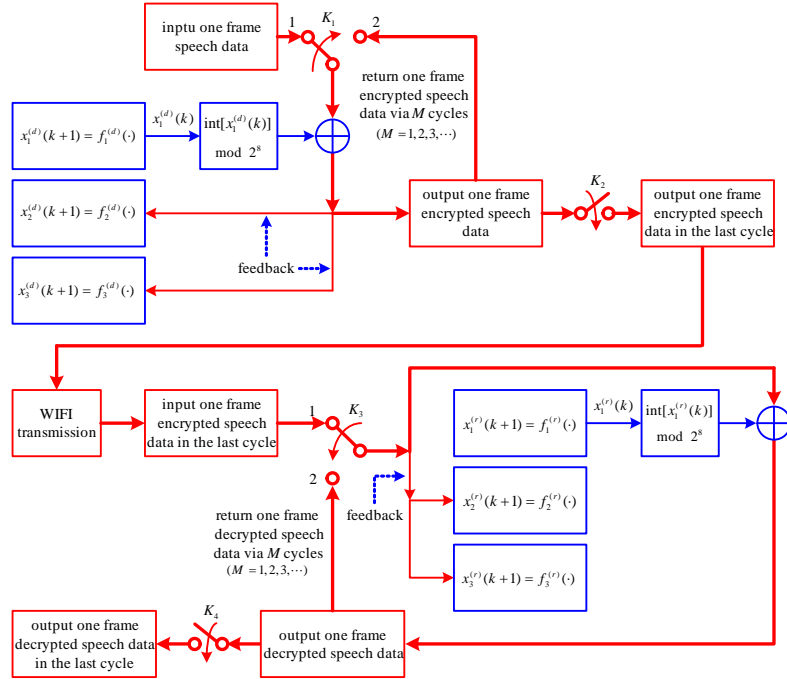


Figure 10: Design principle of multi-round stream cipher.

The basic operation principle of encryption scheme based on multi-round of chaotic stream cipher

shown in Fig. 10 is summarized as follows:

(1) At the sending end, the operation principle of the M -round encryption for each frame of the speech data is as follows: First, place K_1 at position 1 in Fig. 10 and encrypt a frame of signal. Then, place K_1 at position 2 in Fig. 10, the encrypted frame is encrypted again via feedback K_1 . The process is repeated M times. After M -rounds of encryption, K_2 is turned on and this frame of the speech signal at the final round encryption is outputted by the channel. Then, K_1 is located at position 1 again, the next frame of the speech signal is encrypted similarly. By this way, the M -rounds of encryption is done frame by frame.

(2) At the receiving end, the operating principle of the M rounds of decryption for each frame of the speech data is illustrated as follows: First, K_3 is located at position 1, a frame of signal is decrypted. Then K_3 is located at position 2, this frame of signal is decrypted again by feedback K_3 . The process is repeated M times. After M rounds of encryption, K_4 is turned on and this frame of the speech signal at the final round decryption is outputted. Then, K_3 is located at position 1 again, the next frame of the speech signal is decrypted similarly.

According to Fig. 10, a chaotic system (14) is used for the encryption, the basic iterative equation of chaotic map-based encryption system at the sending end is

$$\begin{cases} x_1^{(d)}(k+1) = a_{11}^{(d)} x_1^d(k) + a_{12}^{(d)} x_2^r(k) + a_{13}^{(d)} x_3^d(k), \\ x_2^{(d)}(k+1) = a_{21}^{(d)} p(k) + a_{22}^{(d)} x_2^d(k) + a_{23}^{(d)} x_3^d(k), \\ x_3^{(d)}(k+1) = a_{31}^{(d)} p(k) + a_{32}^{(d)} x_2^d(k) + a_{33}^{(d)} x_3^d(k) \\ \quad + \varepsilon^{(d)} \sin(\sigma^{(d)} p(k)). \end{cases} \quad (15)$$

At the receiver, one gets the decryption system, given by

$$\begin{cases} x_1^{(r)}(k+1) = a_{11}^{(r)} x_1^r(k) + a_{12}^{(r)} x_2^r(k) + a_{13}^{(r)} x_3^r(k), \\ x_2^{(r)}(k+1) = a_{21}^{(r)} p(k) + a_{22}^{(r)} x_2^r(k) + a_{23}^{(r)} x_3^r(k), \\ x_3^{(r)}(k+1) = a_{31}^{(r)} p(k) + a_{32}^{(r)} x_2^r(k) + a_{33}^{(r)} x_3^r(k) \\ \quad + \varepsilon^{(r)} \sin(\sigma^{(r)} p(k)), \end{cases} \quad (16)$$

where $p(k) = \text{mod}(\lfloor x_1^{(d)}(k) \rfloor, 2^8) \oplus s(k)$, $s(k)$ is the k -th element of the input speech signal, and \oplus denotes exclusive OR operation.

When parameters of the both sides match, one has $a_{i,j}^{(d)} = a_{i,j}^{(r)} = a_{i,j}$, ($1 \leq i, j \leq 3$), $\varepsilon^{(d)} = \varepsilon^{(r)} = \varepsilon$, and $\sigma^{(d)} = \sigma^{(r)} = \sigma$. First, according to the second and third equations of system (15) and (16), the iterative equation corresponding to the error signal is

$$\begin{pmatrix} \Delta x_2(k+1) \\ \Delta x_3(k+1) \end{pmatrix} = \begin{pmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{pmatrix} \cdot \begin{pmatrix} \Delta x_2(k) \\ \Delta x_3(k) \end{pmatrix}.$$

Then, one can further get

$$\begin{pmatrix} \Delta x_2(k) \\ \Delta x_3(k) \end{pmatrix} = \begin{pmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{pmatrix}^k \cdot \begin{pmatrix} \Delta x_2(0) \\ \Delta x_3(0) \end{pmatrix}, \quad (17)$$

where $\Delta x_2(0)$ and $\Delta x_3(0)$ are initial values, $\Delta x_2 = x_2^{(r)}(k) - x_2^{(d)}(k)$, and $\Delta x_3 = x_3^{(r)}(k) - x_3^{(d)}(k)$. Taking norm on both sides of Eq. (17), one can get

$$\left\| \begin{pmatrix} \Delta x_2(k) \\ \Delta x_3(k) \end{pmatrix} \right\| \leq \left\| \begin{pmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{pmatrix} \right\|^k \cdot \left\| \begin{pmatrix} \Delta x_2(0) \\ \Delta x_3(0) \end{pmatrix} \right\|. \quad (18)$$

As the characteristic roots corresponding to $\begin{pmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{pmatrix}$ located in the unit circle, one can assure that its norm is less than one from the Gershgorin circle theorem. So, one has

$$\lim_{k \rightarrow \infty} \|\Delta x_i(k)\| = \lim_{k \rightarrow \infty} \|x_i^{(r)}(k) - x_i^{(d)}(k)\| = 0, \quad (19)$$

where $i = 2, 3$.

Now, we further consider the convergence of the error variable $\Delta x_1(k)$. According to the first equation of system (15) and (16), one can get error iterative equation as

$$\Delta x_1(k+1) = a_{11}\Delta x_1(k) + a_{12}\Delta x_2(k) + a_{13}\Delta x_3(k), \quad (20)$$

where $\Delta x_1(k) = x_1^{(r)}(k) - x_1^{(d)}(k)$.

When $k \rightarrow \infty$, one can get $\Delta x_2(k) \rightarrow 0$ and $\Delta x_3(k) \rightarrow 0$ from Eq. (19). So, one has

$$\lim_{k \rightarrow \infty} \Delta x_1(k+1) = \lim_{k \rightarrow \infty} a_{11}\Delta x_1(k). \quad (21)$$

As $|a_{11}| < 1$, one has

$$\lim_{k \rightarrow \infty} \|\Delta x_1\| = \lim_{k \rightarrow \infty} \|x_1^{(r)}(k) - x_1^{(d)}(k)\| = 0. \quad (22)$$

Now, one can see that Eq. (15) and (16) can be synchronized when the parameters are matched. In the receiving end, the original speech signal can be correctly decrypted as $\hat{s}(k) = s(k) \oplus \text{mod}(\lfloor x_1^{(d)}(k) \rfloor, 2^8) \oplus \text{mod}(\lfloor x_1^{(r)}(k) \rfloor, 2^8) = s(k)$. In addition, with the matched parameters, the case for multiple rounds of encryption has the similar results. Note that Eq. (19) and (22) can assure exponential asymptotic convergence, synchronization errors can be ignored just after a few iterations, and the synchronization of the chaotic system at the sending end and the receiving end can be achieved quickly, i.e., the speech signal can be decrypted exactly.

6 Design of a multicast scheme for multiuser wireless communication and its Hardware implementation

The diagram of the multicast-multiuser chaotic map-based speech communication system is shown in Fig. 11, and the corresponding hardware platform is shown in Fig. 12. At the sending end, after IMA-ADPCM compression of the input speech signal, three-level chaotic encryption scheme on speech communication is exerted. The specific encryption and its corresponding encryption keys are described as follows:

The first stage is that the position of every bit of the speech data is permuted randomly, the encryption key is the arrangement of the matrices given in Eq. (7), i.e.,

$$\mathbf{T}_{12}\mathbf{T}_{13}\mathbf{T}_{14}\mathbf{T}_{15}\mathbf{T}_{16}\mathbf{T}_{23}\mathbf{T}_{24}\mathbf{T}_{25}\mathbf{T}_{26}\mathbf{T}_{34}\mathbf{T}_{35}\mathbf{T}_{36}\mathbf{T}_{45}\mathbf{T}_{46}\mathbf{T}_{56}.$$

The second stage is scrambling of 8-bit data position, the encryption key is the arrangement of the matrix (10), i.e.,

$$\mathbf{T}_{12}\mathbf{T}_{13}\mathbf{T}_{14}\mathbf{T}_{15}\mathbf{T}_{16}\mathbf{T}_{17}\mathbf{T}_{23}\mathbf{T}_{24}\mathbf{T}_{25}\mathbf{T}_{26}\mathbf{T}_{27}\mathbf{T}_{34}\mathbf{T}_{35}\mathbf{T}_{36}\mathbf{T}_{37}\mathbf{T}_{45}\mathbf{T}_{46}\mathbf{T}_{47}\mathbf{T}_{56}\mathbf{T}_{57}\mathbf{T}_{67}.$$

The third stage is encrypting speech data values by multi-round stream cipher, the encryption key of parameter matrix is

$$\mathbf{A} = \begin{pmatrix} 0.205 & -0.595 & 0.265 \\ -0.265 & -0.125 & 0.595 \\ 0.33 & -0.33 & 0.47 \end{pmatrix}.$$

At the receiving end, the original speech signal can be recovered by the corresponding inverse process only when all the decryption key at the receiving end matches the encryption key at the sending end. In this experiment, we select number of rounds for encryption $M = 5$. The construction of multicast-multiuser includes one server at sending end and two clients at the receiving end (note that this construction of multicast-multiuser is also applicable for multiple clients at receiving end), the hardware experimental results are obtained as follows:

(1) The ARM implementation of the original speech signal input and the speech signal with three-level chaotic encryption after IMA-ADPCM compression are shown in the top and the bottom of Fig. 13a), respectively.

(2) When all the keys for the server at sending end and two clients at receiving end are matched, the ARM implementation results for the original speech signal, the decrypted speech signal at the first receiving end and the decrypted speech signal at the second receiving, are shown in the three rows of Fig. 13b), respectively.

(3) All the keys for the server at sending end and the second client at receiving end are matched, but one decryption key for one round of decryption for the first client at the receiving end mismatches the encryption key at the receiving end. These lead to the following three conditions:

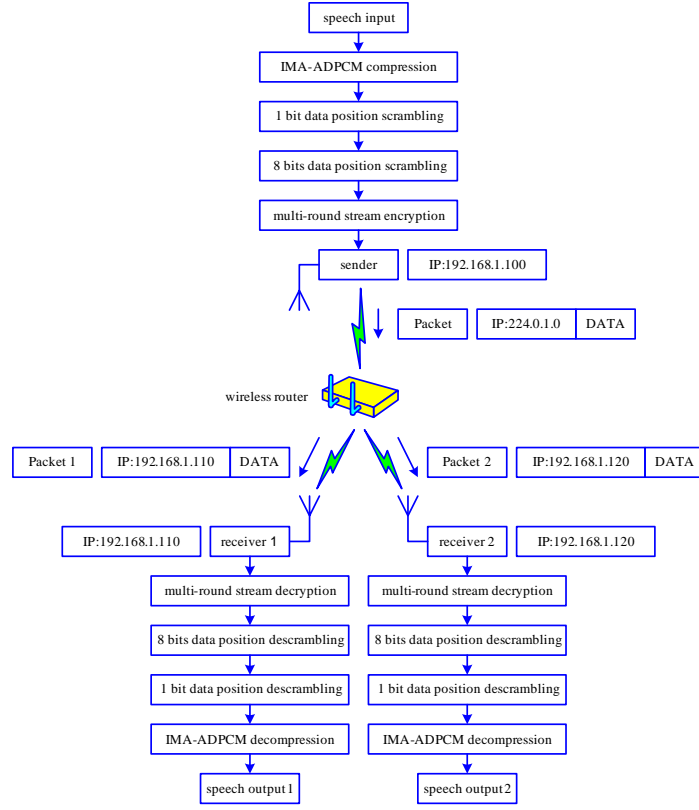


Figure 11: The framework of the multicast-multuser chaotic map-based speech communication system.

1. The first level encryption key for the server at sending end is

$$\mathbf{T}_{12}\mathbf{T}_{13}\mathbf{T}_{14}\mathbf{T}_{15}\mathbf{T}_{16}\mathbf{T}_{23}\mathbf{T}_{24}\mathbf{T}_{25}\mathbf{T}_{26}\mathbf{T}_{34}\mathbf{T}_{35}\mathbf{T}_{36}\mathbf{T}_{45}\mathbf{T}_{46}\mathbf{T}_{56}.$$

Then, the mismatched key of the first level decryption for the first client at receiving end is

$$\mathbf{T}_{34}\mathbf{T}_{13}\mathbf{T}_{14}\mathbf{T}_{15}\mathbf{T}_{16}\mathbf{T}_{23}\mathbf{T}_{24}\mathbf{T}_{25}\mathbf{T}_{26}\mathbf{T}_{12}\mathbf{T}_{35}\mathbf{T}_{36}\mathbf{T}_{45}\mathbf{T}_{46}\mathbf{T}_{56}.$$

2. The second level encryption key for the server at sending end is

$$\mathbf{T}_{12}\mathbf{T}_{13}\mathbf{T}_{14}\mathbf{T}_{15}\mathbf{T}_{16}\mathbf{T}_{17}\mathbf{T}_{23}\mathbf{T}_{24}\mathbf{T}_{25}\mathbf{T}_{26}\mathbf{T}_{27}\mathbf{T}_{34}\mathbf{T}_{35}\mathbf{T}_{36}\mathbf{T}_{37}\mathbf{T}_{45}\mathbf{T}_{46}\mathbf{T}_{47}\mathbf{T}_{56}\mathbf{T}_{57}\mathbf{T}_{67}.$$

Then, the mismatched key for the second level decryption for the first client at receiving end is

$$\mathbf{T}_{45}\mathbf{T}_{13}\mathbf{T}_{14}\mathbf{T}_{15}\mathbf{T}_{16}\mathbf{T}_{17}\mathbf{T}_{23}\mathbf{T}_{24}\mathbf{T}_{25}\mathbf{T}_{26}\mathbf{T}_{27}\mathbf{T}_{34}\mathbf{T}_{35}\mathbf{T}_{36}\mathbf{T}_{37}\mathbf{T}_{12}\mathbf{T}_{46}\mathbf{T}_{47}\mathbf{T}_{56}\mathbf{T}_{57}\mathbf{T}_{67}.$$

3. The key for the third level encryption for the server at sending end is

$$\mathbf{A} = \begin{pmatrix} 0.205 & -0.595 & 0.265 \\ -0.265 & -0.125 & 0.595 \\ 0.33 & -0.33 & 0.47 \end{pmatrix}.$$

Then, the mismatched key for the third level decryption for the first client at receiving end is $\Delta a_{11} = 10^{-9}$ instead of a_{11} . When there is any given mismatched key at any level of decryption, the ARM implementation results for the original speech signal, the decrypted speech signal at the first receiving end and the decrypted speech signal at the second receiving, are obtained in the three rows of Fig. 13c), respectively.

(4) All the keys for the server at sending end and the first client at receiving end are matched, but one decryption key for one round of decryption for the second client at the receiving end mismatches the

encryption key at the receiving end. These cause the similar three conditions as above. Then, the ARM implementation results for the original speech signal, the decrypted speech signal at the first receiving end and the decrypted speech signal at the second receiving end, are obtained in the three rows of Fig. 13d), respectively. The video recordings of the ARM-embedded hardware experimental results can be accessed in the companion page of [26].

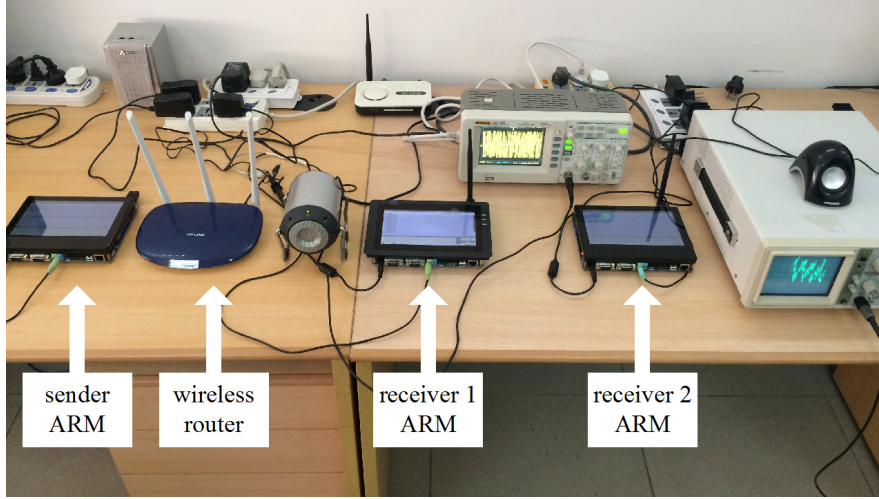


Figure 12: The hardware implementation of the multicast-multiuser chaotic map-based speech communication system.

7 Analysis of security performance

7.1 Test results with NIST test suite

In the test suite designed by National Institute of Standards and Technology in [29], 100 sequences of length 10^6 bits, the P -value fall in the interval $[0.0001, 1]$, it is considered that the random bit sequence generator pass the test, and fail otherwise.

In the encryption scheme composed of multi-round stream cipher shown in Fig. 10, after the iterative sequence is converted into 64-bit integers, its eight least significant bits of the speech signal are encrypted. Therefore, the 1×10^8 bits sequence is made up of every eight least significant bits of iterative sequence in the corresponding NIST test suite, the test results are shown in Table 1, where symbol “-” indicates that the corresponding test failed. The relationship between the test results and the Lyapunov exponents is explained as follows:

(1) The results of P-value (1) with the corresponding parameter $\varepsilon = 100$ and $\sigma = 0.0667$, the Lyapunov exponents are obtained as $LE_1 = -0.0017$, $LE_2 = -0.0993$, $LE_3 = -0.2229$, respectively. Since the Lyapunov exponents are all negative, the generated sequence can not pass any item in the NIST test suite.

(2) The results of P-value (2) with the corresponding parameter $\varepsilon = 300$ and $\sigma = 0.2$, the Lyapunov exponents are obtained as $LE_1 = 1.1041$, $LE_2 = 0.9588$, $LE_3 = 0.1828$, respectively. Since the Lyapunov exponents are not big enough, only two items in the NIST test suite can be passed.

(3) The results of P-value (3) with the corresponding parameter $\varepsilon = 3000$ and $\sigma = 2$, the Lyapunov exponents are obtained as $LE_1 = 3.3825$, $LE_2 = 3.287$, $LE_3 = 0.1893$, respectively. Only nine items in the NIST test suite can be passed.

(4) The results of P-value (4) with the corresponding parameter $\varepsilon = 30000$ and $\sigma = 2$, the Lyapunov exponents are obtained as $LE_1 = 5.6879$, $LE_2 = 5.65$, $LE_3 = 0.1895$, respectively. Only 13 items in the NIST test suite can be passed.

(5) The results of P-value (5) with the corresponding parameter $\varepsilon = 3 \cdot 10^5$ and $\sigma = 200$, the Lyapunov exponents are obtained as $LE_1 = 7.9827$, $LE_2 = 7.9814$, $LE_3 = 0.1895$, respectively. In this case, fourteen items in the NIST test suite can be passed.

(6) The results of P-value (6) with the corresponding parameter $\varepsilon = 3 \cdot 10^8$ and $\sigma = 2 \cdot 10^5$, the Lyapunov exponents are obtained as $LE_1 = 14.8403$, $LE_2 = 14.8107$, $LE_3 = 0.1903$, respectively. Since the Lyapunov exponents are all positive and big enough, all items in the NIST test suite can be passed.

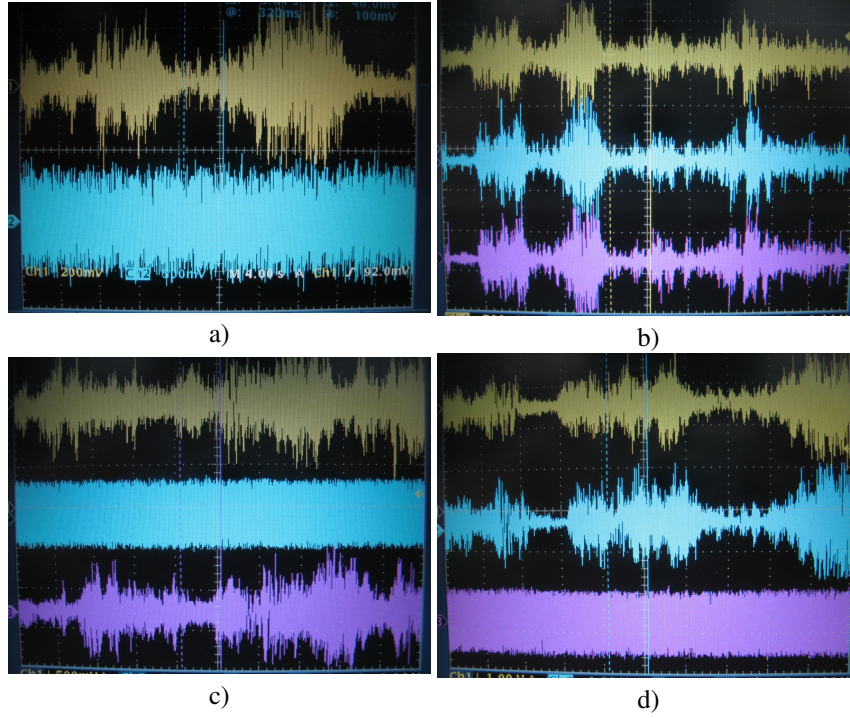


Figure 13: The original speech signal and the encrypted speech signal.

Table 1: The performed tests and the ratio of sequences passing each test in a sample of 100 sequences.

Statistical Test	P-value (1)	P-value (2)	P-value (3)	P-value (4)	P-value (5)	P-value (6)
Frequency	—	—	0.946308	0.867692	0.798139	0.759756
Block Frequency	—	—	0.035174	0.062821	0.574903	0.437274
Cumulative Sums	—	—	0.7886635	0.448529	0.67685	0.2636575
Runs	—	—	0.574903	0.12962	0.035174	0.616305
Long Runs of Ones	—	—	0.455937	0.249284	0.090936	0.534146
Rank	—	0.108791	0.289667	0.946308	0.181557	0.946308
FFT	—	—	0.514124	0.534146	0.699313	0.096578
Non-overlapping Templates	—	—	—	—	—	0.528973
Overlapping Templates	—	—	0.249284	0.816537	0.924076	0.191687
Universal	—	—	0.867692	0.55442	0.181557	0.474986
Approximate Entropy	—	—	—	0.319084	0.595549	0.739918
Random Excursions	—	—	—	0.3943705	0.35054825	0.68632925
Random Excursions Variant	—	—	—	—	0.344756	0.6600216
Serial	—	—	—	0.428899	0.0458625	0.7494675
Linear Complexity	—	0.366918	—	0.991468	0.935716	0.032923
Success	0/15	2/15	9/15	13/15	14/15	15/15

As shown in Table 1, the value of positive Lyyapunov exponents increases and degeneracy disappears as increase of ε and σ , so the generated sequence can pass all test items.

7.2 Statistical analysis

Taking one frame of speech data for example, the statistical analysis of this frame of speech data before and after the encryption by multi-round stream cipher is performed to obtain the corresponding histogram shown in Fig. 14.

As presented in Fig. 14, the chaotic sequence owns good randomness, so the encrypted audio data approaches uniform distribution. Thus, the proposed encryption method makes the ciphertext own weak correlation with the secrete key,

$$H(K|C_1C_2\cdots,C_n) \approx H(K),$$

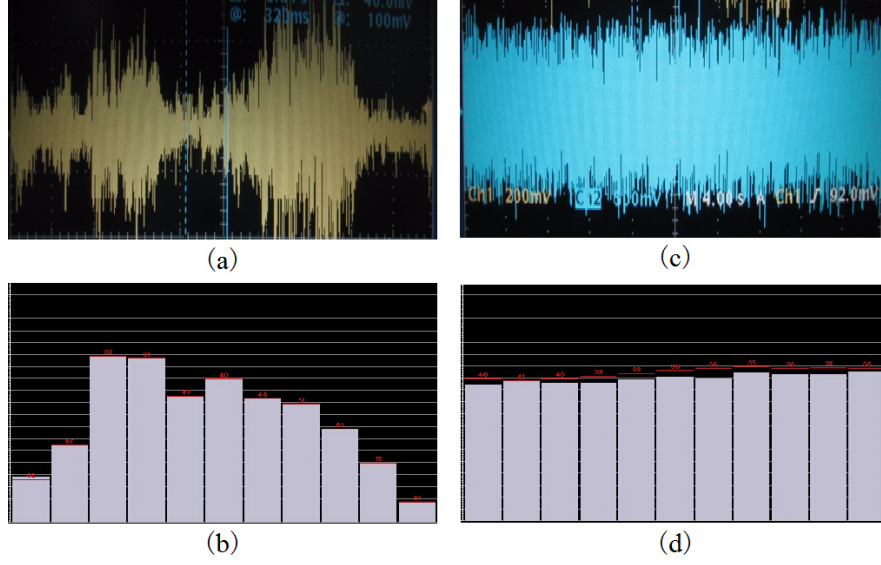


Figure 14: The statistical analysis of one frame of speech data before and after the encryption by multi-round stream cipher: (a) The original speech signal; (b) The distribution of the original speech signal; (c) The speech signal encrypted by multi-round stream cipher; (d) The distribution of the speech signal encrypted by multi-round stream cipher.

where H is the information entropy, K is the secret key, $C_1C_2\cdots,C_n$ are cipher elements. Thus, it increases the capability of withstanding statistical attacks and ciphertext-only attack much.

7.3 Analysis of performance against differential attack

The differential analysis plays an important role in the security analysis of chaotic cryptosystem, it can demonstrate whether the encryption results are sensitive to change of the plaintext, namely the “avalanche effect”. If the chaotic cryptosystem can withstand the differential attack, any variation in the plaintext could cause serious affect in the ciphertext. Such impact is widely measured by tow factors, NPCR (Number of Pixels Change Rate) and UACI (Unified Average Changing Intensity) [2],

$$\begin{cases} \text{NPCR} = \frac{\sum_i D(i)}{F} \cdot 100\%, \\ \text{UACI} = \frac{1}{F} \cdot \sum_i \frac{|C(i) - C'(i)|}{255} \cdot 100\%, \end{cases} \quad (23)$$

where

$$\begin{cases} C = [C_1C_2\cdots C_F] \\ C' = [C'_1C'_2\cdots C'_F], \end{cases}$$

$$D(i) = \begin{cases} 1 & \text{if } C(i) \neq C'(i), \\ 0 & \text{if } C(i) = C'(i), \end{cases}$$

$C(i)$ and $C'(i)$ denote the i -th element of two ciphertext C and C' , respectively.

To study influence of each element of the original speech data for the corresponding ciphertext, a frame of speech data of 16384 elements is selected to calculate NPCR and UACI corresponding to each element, then the average value is calculated as

$$\begin{cases} \overline{\text{NPCR}} = \sum_{m=1}^{16384} \text{NPCR}(m)/16384, \\ \overline{\text{UACI}} = \sum_{m=1}^{16384} \text{UACI}(m)/16384, \end{cases} \quad (24)$$

where $\text{NPCR}(m)$ and $\text{UACI}(m)$ denote the NPCR value and UACI value of the m -th element.

According to Eq. (24), it is shown in Fig. 10 that $\overline{\text{NPCR}} = 49.8\%$ and $\overline{\text{UACI}} = 16.73\%$ after first round of encryption. When $M = 2$, it is shown in Fig. 10 that $\overline{\text{NPCR}} = 99.61\%$ and $\overline{\text{UACI}} = 33.47\%$ after second round encryption. In addition, NPCR and UACI become more and more larger with increase of round number.

By comparison of the above results, it is obvious that the ability of one-round encryption to resist differential attack is weaker than that of two-round encryption. The underlying mechanisms are detailed below.

In the method shown in Fig. 10, the sequence is encrypted from left to right in the encryption of one frame of speech data by one-round chaotic stream cipher, encryption for one speech frame is arranged as $s(1) \rightarrow s(2) \rightarrow s(3), \dots, s(16383) \rightarrow s(16384)$. Then, after 16384 iterations, the one-round encryption for one frame of speech data is completed.

Figure 10 shows the encryption and decryption process based on chaotic stream cipher, whose main feature is the use of a closed loop feedback system. That means the encrypted results should be fed back to the original one through the feedback system after each encryption, and makes the speech data and chaotic system “all rolled into one”. Thus, any slight variation in the plaintext data has a significant impact on the ciphertext.

However, there is a problem with the one-round encryption scheme: there are obvious differences among the impacts on the ciphertext caused by the variation in the different positions of the plaintext data. Specifically, the variation in data of the smaller index i has stronger influence on the ciphertext, and the calculated NPCR and UACI are relatively big. The underlying reason is that the element of smaller index i is encrypted earlier, the encrypted result can be earlier feedback to the original system through the feedback loop. More iterations the feedback to the original system are, more easier the data and the chaotic system combine. Thus the impacts on the ciphertext caused by the variation in the position are greater. The element of bigger index is encrypted later, and the encrypted results are feedback to the original system later. The more less the feedback to the original system iterates, the more harder efficient combination of the data and the chaotic system becomes. Thus, the impacts on the ciphertext caused by the variation in the position are smaller.

If multi-round encryption scheme is used, the results are different from one-round encryption. For example, in the case of more than two or two rounds encryption scheme, even for speech data $s(16384)$, the encrypted results also have the opportunity to mix efficiently with chaotic system in the next round of the encryption through the feedback loop to the original system. It is a fundamental solution for the problem that the impacts on the ciphertext caused by the variation in the bigger index are small.

7.4 Sensitivity with respect to change of parameters of the discrete chaotic system

Any secure encryption system is expected to own “avalanche effect” with respect to change of secret key, i.e. tiny change of one sub-key can cause dramatic change of the corresponding ciphertext. The smaller magnitude of the mismatch errors, the better security of the system. In this subsection, we analyzed parameter sensitivity of the proposed system under different round numbers.

7.4.1 The case under round number of one

Set the absolute value of error caused by the mismatched key parameters between the receiving end and the sending end as $|\Delta a_{i,j}| = |a_{i,j}^{(r)} - a_{i,j}^{(d)}|$, $1 \leq i, j \leq 3$. Suppose that there are only one mismatched key parameter in $|\Delta a_{i,j}|$ and others are matched. As the original speech signal cannot be decrypted, it was tested that the error matrix caused by one-round encryption scheme shown in Fig. 10 with mismatched key parameter is

$$|\Delta a|_{C1} \triangleq \begin{pmatrix} |\Delta a|_{11} & |\Delta a|_{12} & |\Delta a|_{13} \\ |\Delta a|_{21} & |\Delta a|_{22} & |\Delta a|_{23} \\ |\Delta a|_{31} & |\Delta a|_{32} & |\Delta a|_{33} \end{pmatrix} \propto \begin{pmatrix} 10^{-5} & 10^{-5} & 10^{-5} \\ \underline{5} & 10^{-5} & 10^{-5} \\ \underline{5} & 10^{-5} & 10^{-5} \end{pmatrix}. \quad (25)$$

The underlined part of the above formula indicates that the corresponding key parameter is not sensitive to the mismatch errors, which is mainly caused by that each encrypted speech data $p(k) = \text{mod}([x_1^{(d)}(k)], 2^8) \oplus s(k)$ is fed back to these items by the feedback loop.

According to Eq. (25), one-round encryption scheme shown in Fig. 10 exists two invalid key parameters and only seven valid key parameters, the absolute value of the error for mismatch valid key parameters is of magnitude 10^{-5} . The underlying reasons are analyzed as follows:

According to Eq. (25), the mismatch error of $|\Delta a_{21}|$ and $|\Delta a_{31}|$ are of the same magnitude, the other mismatch errors Δa_{ij} ($i \neq 2, 3; j \neq 1$) are also in the another same magnitude. Therefore, only need to study $|\Delta a_{31}|$ and $|\Delta a_{32}|$, the similar results of other mismatched error can be obtained.

(1) Set $\varepsilon^{(d)} = \varepsilon^{(r)} = \varepsilon$, $\sigma^{(d)} = \sigma^{(r)} = \sigma$, assume that there is only one mismatch error $|\Delta a_{31} = a_{31}^{(r)} - a_{31}^{(d)}|$ in a_{31} , and the other parameters are matched, according to Eq. (15) and (16), the iterative equation of the error signal is given as

$$\begin{cases} \Delta x_1(k+1) = a_{11}\Delta x_1(k) + a_{12}\Delta x_2(k) + a_{13}\Delta x_3(k), \\ \Delta x_2(k+1) = a_{22}\Delta x_2(k) + a_{23}\Delta x_3(k), \\ \Delta x_3(k+1) = \Delta a_{31}p(k) + a_{32}\Delta x_2(k) + a_{33}\Delta x_3(k). \end{cases} \quad (26)$$

When the error signal reaches a steady state, $\Delta x_i(k+1) = \Delta x_i(k)$ is satisfied for $i = 1, 2, 3$. The above equation becomes

$$\begin{cases} \Delta x_1(k) = a_{11}\Delta x_1(k) + a_{12}\Delta x_2(k) + a_{13}\Delta x_3(k), \\ \Delta x_2(k) = a_{22}\Delta x_2(k) + a_{23}\Delta x_3(k), \\ \Delta x_3(k) = \Delta a_{31}p(k) + a_{32}\Delta x_2(k) + a_{33}\Delta x_3(k). \end{cases} \quad (27)$$

Solving the above equation, one can obtain

$$\begin{cases} \Delta x_1(k) = -0.0887 \cdot \Delta a_{31}p(k), \\ \Delta x_2(k) = 0.7507 \cdot \Delta a_{31}p(k), \\ \Delta x_3(k) = 1.4194 \cdot \Delta a_{31}p(k). \end{cases} \quad (28)$$

(2) Set $\varepsilon^{(d)} = \varepsilon^{(r)} = \varepsilon$, $\sigma^{(d)} = \sigma^{(r)} = \sigma$, assume that there is only one mismatch error $|\Delta a_{32} = a_{32}^{(r)} - a_{32}^{(d)}|$ in a_{32} , and the other parameters are matched, $a_{ij}^{(r)} = a_{ij}^{(d)} = a_{ij}$ for $i \neq 3, j \neq 2$. According to Eq. (15) and (16), the iterative equation with an error signal is given as

$$\begin{cases} \overline{\Delta x_1}(k+1) = a_{11}\overline{\Delta x_1}(k) + a_{12}\overline{\Delta x_2}(k) + a_{13}\overline{\Delta x_3}(k), \\ \overline{\Delta x_2}(k+1) = a_{22}\overline{\Delta x_2}(k) + a_{23}\overline{\Delta x_3}(k), \\ \overline{\Delta x_3}(k+1) = a_{32}\overline{\Delta x_2}(k) + \Delta a_{32}x_2(k) + a_{33}\overline{\Delta x_3}(k). \end{cases} \quad (29)$$

When the error signal reaches a steady state, $\overline{\Delta x_i}(k+1) = \overline{\Delta x_i}(k)$ is satisfied for $i = 1, 2, 3$. The above equation becomes

$$\begin{cases} \overline{\Delta x_1}(k) = a_{11}\overline{\Delta x_1}(k) + a_{12}\overline{\Delta x_2}(k) + a_{13}\overline{\Delta x_3}(k), \\ \overline{\Delta x_2}(k) = a_{22}\overline{\Delta x_2}(k) + a_{23}\overline{\Delta x_3}(k), \\ \overline{\Delta x_3}(k) = a_{32}\overline{\Delta x_2}(k) + \Delta a_{32}x_2(k) + a_{33}\overline{\Delta x_3}(k). \end{cases} \quad (30)$$

Solving the above set of equations, one can obtain

$$\begin{cases} \overline{\Delta x_1}(k) = -0.0887 \cdot \Delta a_{32}x_2(k), \\ \overline{\Delta x_2}(k) = 0.7507 \cdot \Delta a_{32}x_2(k), \\ \overline{\Delta x_3}(k) = 1.4194 \cdot \Delta a_{32}x_2(k). \end{cases} \quad (31)$$

According to Fig. 9, the maximum magnitude of $x_2(k)$ is 10^8 . From Fig. 10, one can see that the maximum magnitude of $p(k)$ is 2^8 . Dividing the left part and the right part of Eq. (28) with that of Eq. (31), respectively, one can get

$$\frac{|\Delta a_{31}|}{|\Delta a_{32}|} \propto \frac{|x_2(k)|_{\max}}{|p(k)|_{\max}} \propto 10^5.$$

This explains insensitivity of parameter mismatch and error matrix given in Eq. (25).

7.4.2 The case under multiple rounds

In multi-round of encryption, for example, set $M = 2$ in Fig. 10, it was tested that the error matrix caused by two-round encryption in Fig. 10 with mismatched key parameter is

$$|\Delta a|_{C2} \propto \begin{pmatrix} 10^{-8} & 10^{-8} & 10^{-8} \\ 10^{-2} & 10^{-8} & 10^{-8} \\ 10^{-2} & 10^{-8} & 10^{-8} \end{pmatrix}. \quad (32)$$

According to the above equation, two-round encryption in Fig. 10 exists 9 key parameters, the errors of mismatched key parameters are of magnitude between 10^{-8} and 10^{-2} . Similarly, set $M = 5$ in Fig. 10, it was tested that the error matrix caused by five-rounds encryption in Fig. 10 with mismatched key parameter is

$$|\Delta a|_{C5} \propto \begin{pmatrix} 10^{-9} & 10^{-9} & 10^{-9} \\ 10^{-3} & 10^{-9} & 10^{-9} \\ 10^{-3} & 10^{-9} & 10^{-9} \end{pmatrix}. \quad (33)$$

In particular, the absolute value of error caused by the mismatched key parameters in Eq. (25), (32), (33) does not need to be very accurate, and the coarse scope is enough. In addition, Eq. (25) and (32), (33) mean that receiving end can't decrypt the original speech signal if error of any key parameter is greater than or equal to a given magnitude. In conclusion, by comparing Eq. (25), (32), (33), we can conclude that the avalanche effect in the mismatch of key parameter is significantly enhanced with increase of the round number.

When the round number of encryption is increased, the key parameter becomes increasingly sensitive due to the extreme sensitivity of the chaotic system, and with increase of the number and quantity of positive Lyapunov exponents. In Fig. 10, at the receiving end, as long as there is a mismatched parameter, every parameter error after each round of encryption leads to error diffusion. But the proposed hyper chaotic system does not have degeneration, and the positive Lyapunov exponents are sufficiently large, thus the sensitivity of this parameter error is greater, and error diffusion is stronger.

7.5 Estimation of the space of secret key

According to the order of multiplication in each transformation sub-matrix \mathbf{T}_{ij} of Eq. (2), one has

$$\mathbf{A} = \prod_{i=1}^{n-1} \left(\prod_{j=i+1}^n \mathbf{T}_{\alpha_i \beta_j} \right) \quad (34)$$

by ranking the subscript of \mathbf{T}_{ij} with all possible orders, where α_i and β_j should satisfy the following conditions:

$$\begin{cases} \alpha_i \in \{1, 2, \dots, n-1\}, \\ \alpha_p \neq \alpha_q \text{ if } p \neq q, \\ \beta_j \in \{2, 3, \dots, n\} \text{ for } j = i+1, i+2, \dots, n, \\ \beta_r \neq \beta_s \text{ if } r \neq s, \\ \beta_j > \alpha_i. \end{cases} \quad (35)$$

According to the above rules, the subscript of $\mathbf{T}_{\alpha_i \beta_j}$ in (34) is sorted as

$$S = \alpha_1 \beta_2, \alpha_1 \beta_3, \dots, \alpha_1 \beta_n, \alpha_2 \beta_3, \alpha_2 \beta_4, \dots, \alpha_2 \beta_n, \dots, \alpha_{n-1} \beta_n.$$

In each exchange of the subscript $\alpha_i \beta_j$ of $\mathbf{T}_{\alpha_i \beta_j}$, we get a corresponding new subscript order, each of which can be used as a key for the speech encryption. According to the above ordering rule, the number of all possible orders is $K_s^{(1)} = \binom{n}{2}$. Then according to order of priority $E(S, \alpha_i)$ in Eq. (5), the total corresponding sort is $K_s^{(2)} = n!$.

(1) In the first level of encryption, position of every bit of the compressed data is permuted randomly, set $n = 6$, the number of possible order is

$$K_s^1 = K_s^{(1)} \cdot K_s^{(2)} = \binom{6}{2} \cdot 6! = 9.4154 \cdot 10^{14}.$$

(2) In the second level of encryption, the obtained data is further permuted in the level of byte, set $n = 7$, the number of all possible order is

$$K_s^2 = K_s^{(1)} \cdot K_s^{(2)} = \binom{7}{2} \cdot 7! = 2.575 \cdot 10^{23}.$$

(3) In the third level of encryption, it is operated with a multi-round chaotic stream cipher. At 5-rounds encryption, the number of the valid key parameters is 9. When the error for mismatch valid key parameters $\{\Delta a_{i,j}\}_{i=1,j=1}^{3,3}$ is less than the value in the error matrix mismatch valid key parameters in Eq. (33). The original speech signal can be decrypted. Therefore, the size of key space is

$$K_s^3 = \prod_{i=1}^3 \left(\prod_{j=1}^3 \frac{|a_{i,j}|}{|\Delta a_{i,j}|} \right) = 3.2607 \cdot 10^{64}.$$

To sum up, the size of the key space for three-level encryption system is

$$K_s = K_s^1 \cdot K_s^2 \cdot K_s^3 = 7.9055 \cdot 10^{102}.$$

8 Conclusion

In this paper, a chaotic map-based multicast scheme for multiuser wireless communication was designed and implemented in an ARM-embedded platform. Detailed hardware experiment results are provided to show its real performance. Comprehensive theoretical analysis results support security performance of the proposed secure speech wireless communication scheme.

Acknowledgement

This work was supported by the National Key Research and Development Program of China (No. 2016YFB0800401), the National Natural Science Foundation of China (No. 61671161, 61532020, 11072254), Science and Technology Planning Project of Guangzhou (No. 20151001036), and Hunan Provincial Natural Science Foundation of China (no. 2015JJ1013).

References

- [1] Liu K, Zhu H, Lü J. Bridging the gap between transmission noise and sampled data for robust consensus of multi-agent systems. *IEEE Transactions on Circuits and Systems I: Regular Papers* 2015; **62**(7):1836–1844.
- [2] Yu S, Lü J, Chen G. *Chaotification of Dynamical Systems and Its Applications*. Science Press: Beijing, China, 2013.
- [3] Ren HP, Baptista MS, Grebogi C. Wireless communication with chaos. *Physical Review Letters* 2013; **110**(18):184 101.
- [4] Li C. Cracking a hierarchical chaotic image encryption algorithm based on permutation. *Signal Processing* 2016; **118**:203–210.
- [5] Xie EY, Li C, Yu S, Lü J. On the cryptanalysis of Fridrich's chaotic image encryption scheme. *Signal Processing* 2017; **132**, doi:http://dx.doi.org/10.1016/j.sigpro.2016.10.002.
- [6] Li C, Liu Y, Xie T, Chen MZQ. Breaking a novel image encryption scheme based on improved hyper-chaotic sequences. *Nonlinear Dynamics* 2013; **73**(3):2083–2089.
- [7] Tong XJ, Wang Z, Zhang M, Liu Y, Xu H, Ma J. An image encryption algorithm based on the perturbed high-dimensional chaotic map. *Nonlinear Dynamics* 2015; **80**(3):1493–1508.
- [8] Celikovskiy S, Chen G. Secure synchronization of a class of chaotic systems from a nonlinear observer approach. *IEEE Transactions on Automatic Control* 2005; **50**(1):76–82.

- [9] Sheu LJ. A speech encryption using fractional chaotic systems. *Nonlinear Dynamics* 2011; **65**(1–2):103–108.
- [10] Liu K, Wu L, Lü J, Zhu H. Finite-time adaptive consensus of a class of multi-agent systems. *Science China Technological Sciences* 2016; **59**(1):22–32.
- [11] Bose R. Novel public key encryption technique based on multiple chaotic systems. *Physical Review Letters* 2005; **95**:art. no. 098 702.
- [12] Li S, Li C, Lo KT, Chen G. Cryptanalyzing an encryption scheme based on blind source separation. *IEEE Transactions on Circuits and Systems I-Regular Papers* 2008; **55**(4):1055–1063.
- [13] Zhu Y, Lu Y, Vikram A. On privacy of encrypted speech communications. *IEEE Transactions on Dependable and Secure Computing* 2012; **9**(4):470–481.
- [14] Yu S, Tang WK, Lü J, Chen G. Generating 2n-wing attractors from lorenz-like systems. *International Journal of Circuit Theory and Applications* 2010; **38**(3):243–258.
- [15] Shen C, Yu S, Lü J, Chen G. Constructing hyperchaotic systems at will. *International Journal of Circuit Theory and Applications* 2015; **43**(12):2039–2056.
- [16] Wang Q, Yu S, Li C, Lü J, Fang X, Guyeux C, Bahi JM. Theoretical design and FPGA-based implementation of higher-dimensional digital chaotic systems. *IEEE Transactions on Circuits and Systems I-Regular Papers* 2016; **63**(3):401–412.
- [17] Tan S, Lü J, Hill DJ. Towards a theoretical framework for analysis and intervention of random drift on general networks. *IEEE Transactions on Automatic Control* 2015; **60**(2):576–581.
- [18] Goldburg B, Sridharan S, Dawson E. Design and cryptanalysis of transform-based analog speech scramblers. *IEEE Journal on Selected Areas in Communications* 1993; **11**(5):735–744.
- [19] Kak S, Jayant N. On speech encryption using waveform scrambling. *Bell System Technical Journal* 1977; **56**(5):781–808.
- [20] Li C, Lo KT. Optimal quantitative cryptanalysis of permutation-only multimedia ciphers against plain-text attacks. *Signal Processing* 2011; **91**(4):949–954.
- [21] Wang H, Hempel M, Peng D, Wang W, Sharif H, Chen HH. Index-based selective audio encryption for wireless multimedia sensor networks. *IEEE Transactions on Multimedia* 2010; **12**(3):215–223.
- [22] Zeng W, Lei S. Efficient frequency domain selective scrambling of digital video. *IEEE Transactions on Multimedia* 2003; **5**(1):118–129.
- [23] Servetti A, Martin JCD. Perception-based partial encryption of compressed speech. *IEEE Transactions on Speech and Audio Processing* 2002; **10**(8):637–643.
- [24] Zhu K, Li C, Asari V, Saupe D. No-reference video quality assessment based on artifact measurement and statistical analysis. *IEEE Transactions on Circuits and Systems for Video Technology* 2015; **25**(4):533–546.
- [25] Wu CP, Kuo CCJ. Design of integrated multimedia compression and encryption systems. *IEEE Transactions on Multimedia* 2005; **7**(5):828–839.
- [26] Lin Z, Yu S, Lu J, Cai S, Chen G. Design and ARM-embedded implementation of a chaotic map-based real-time secure video communication system. *IEEE Transactions on Circuits and Systems for Video Technology* 2015; **25**(7):1203–1216. <http://www.ee.cityu.edu.hk/~gchen/VideoCrypt.htm>.
- [27] Cummiskey P, Jayant NS, Flanagan JL. Adaptive quantization in differential PCM coding of speech. *Bell System Technical Journal* 1973; **52**(7):1105–1118.
- [28] Ferro E, Potorti F. Bluetooth and Wi-Fi wireless protocols: A survey and a comparison. *IEEE Wireless Communications* 2005; **12**(1):12–26.
- [29] Rukhin A, et al. A statistical test suite for random and pseudorandom number generators for cryptographic applications. NIST Special Publication 800-22rev1a 2010. http://csrc.nist.gov/groups/ST/toolkit/rng/documentation_software.html.